Acta Astronautica 95 (2014) 101–110

Contents lists available at ScienceDirect

Acta Astronautica

journal homepage: www.elsevier.com/locate/actaastro

Launch ascent guidance by discrete multi-model predictive control

Alexandre Vachon [a,*], André Desbiens [a], Eric Gagnon [b], Caroline Bérard [c]

[a] *Université Laval, Pavillon Adrien-Pouliot, 1065 avenue de la Médecine, Québec, Canada G1V 0A6*
[b] *Defence R&D Canada – Valcartier, 2459 boulevard Pie-XI Nord, Québec, Canada G3J 1X5*
[c] *Institut Supérieur de l'Aéronautique et de l'Espace, 10 avenue Édouard-Belin BP 54032, 31055 Toulouse Cedex 4, France*

ARTICLE INFO

Article history:
Received 15 January 2013
Received in revised form
28 September 2013
Accepted 30 October 2013
Available online 8 November 2013

Keywords:
Model predictive control
Linear-time-varying representation
Linear-fractional representation
Guidance
Space launcher

ABSTRACT

This paper studies the application of discrete multi-model predictive control as a trajectory tracking guidance law for a space launcher. Two different algorithms are developed, each one based on a different representation of launcher translation dynamics. These representations are based on an interpolation of the linear approximation of nonlinear pseudo-five degrees of freedom equations of translation around an elliptical Earth. The interpolation gives a linear-time-varying representation and a linear-fractional representation. They are used as the predictive model of multi-model predictive controllers. The controlled variables are the orbital parameters, and constraints on a terminal region for the minimal accepted precision are also included. Use of orbital parameters as the controlled variables allows for a partial definition of the trajectory. Constraints can also be included in multi-model predictive control to reduce the number of unknowns of the problem by defining input shaping constraints. The guidance algorithms are tested in nominal conditions and off-nominal conditions with uncertainties on the thrust. The results are compared to those of a similar formulation with a nonlinear model predictive controller and to a guidance method based on the resolution of a simplified version of the two-point boundary value problem. In nominal conditions, the model predictive controllers are more precise and produce a more optimal trajectory but are longer to compute than the two-point boundary solution. Moreover, in presence of uncertainties, developed algorithms exhibit poor robustness properties. The multi-model predictive control algorithms do not reach the desired orbit while the nonlinear model predictive control algorithm still converges but produces larger maneuvers than the other method.

1. Introduction

It is well known that there are two approaches to guide a vehicle toward its final destination [1]: predictor/corrector methods and path reference methods. Launch ascent guidance of a space launcher is no different than any other vehicle, and both approaches have been applied to it in the past [2]. The predictor/corrector approaches consist in the generation of a new trajectory and the corresponding steering commands at each iteration where the current state is the initial state of a two-point boundary value problem. Hence, complex optimization algorithms are required to solve the problem, or a hypothesis can be formulated to simplify the problem. For the launch ascent trajectory, two hypotheses on the Earth gravity approximation, proportional to the radius [3] or uniform [4], give an analytical solution of the costate system and eliminate the need for

* Corresponding author. Tel.: +1 418 656 2131x5652.
E-mail addresses: alexandre.vachon.2@ulaval.ca (A. Vachon), Andre.Desbiens@gel.ulaval.ca (A. Desbiens), Eric.Gagnon@drdc-rddc.gc.ca (E. Gagnon), Caroline.Berard@isae.fr (C. Bérard).

0094-5765/$ - see front matter © 2013 IAA. Published by Elsevier Ltd. All rights reserved.
http://dx.doi.org/10.1016/j.actaastro.2013.10.022

complex algorithms. Both of these hypotheses are well studied and can be considered mature techniques suitable for many launch applications [4]. The path reference methods compute the steering commands needed to follow a predefined trajectory. As presented by Shrivastava et al. [2], many linear control methods with time scheduled values have been studied for launch ascent guidance. Among them, Q-guidance is the most important one [5]. Recent developments in nonlinear control theory boost the path reference approaches as linearization of the nonlinear equations of motion is not required. Nonlinear model predictive control (NMPC) [6], neural networks [7] and sliding mode control [8] are examples of the application of nonlinear control methods to the guidance of a space launcher.

Even if, under nominal conditions, the trajectory produced by the path reference methods is closer to the optimal trajectory than the trajectory obtained by the predictor/corrector methods [2], launch ascent guidance is traditionally achieved by predictor/corrector methods [9]. The main reason for this choice is the robustness of the predictor/corrector methods under non-nominal conditions; however, in the exo-atmospheric portion, space launchers are expected to follow their nominal trajectory quite closely as they evolve in an environment that is practically perturbation-free. Therefore, path reference must be considered for this portion of the launch, mainly for vehicles with well-defined missions and reliable components [2].

This paper focuses on MPC-based path reference approaches. Through integral resolution, continuous time predictive control of Lu [6] restricts tracking to the in-plane ascent trajectory around a spherical Earth. The proposed guidance laws are discrete versions of predictive control. In opposition to continuous predictive control, discrete formulation has a finite number of unknowns and does not require the model to have a special form to obtain the solution. Therefore, the in-plane and out-of-plane ascent problem of a motion around an elliptical Earth can be solved. The discrete model is an Euler approximation of the pseudo-5 degrees of freedom (pseudo-5DoF) equations of motion. The prediction of a NMPC is made directly with these equations of motion while the varying linear representations are the bases of multi-model predictive control (MMPC) algorithms. Vachon et al. [10] apply discrete NMPC to space launcher exo-atmospheric guidance. The varying linear representations are time interpolation (linear-time-varying representation (LTVR) and linear-fractional representation (LFR)) of a set of linear models of the launcher translation dynamics [11].

MMPC algorithms based on LTVR and LFR are well-known approaches used to handle multiple operation regimes [12–15]. That being said, they are mostly implemented as linear MPC where the predictive model varies at each iteration [12] or where multiple linear MPC algorithms are simultaneously solved and a second algorithm weighs the results based on the current operating regime [14]. These formulations are valid for systems operating in multiple operating regimes but where the variations are slow and hence the prediction around a single operating regime is valid. When the operating regime varies inside the prediction horizon, the algorithms are based on minmax optimization to obtain the varying model dynamically [13,15]. This is a good approach when the sequence of the operating regimes is not known *a priori*. In a launcher guidance law, the prediction must be made over multiple operating regimes that are known *a priori*. Hence, identifying the varying linear model using minmax optimization at each time step is not necessary, and pre-identified models can be used. The minmax algorithm is therefore converted into a single minimization.

Section 2 gives an overview of the representations of the launcher translation dynamics obtained by Vachon et al. [11]. Section 3 introduces the NMPC and the two MMPC formulations. The resulting formulations are then implemented and tested in simulations. Section 4 compares the results with a specific predictor/corrector method [3].

2. Equations of motion

2.1. Pseudo-five degrees of freedom equations

As stated by Zipfel [16], a guidance law developed for a pseudo-5DoF can be implemented in a full six-degree simulator with no modifications. Pseudo-5DoF equations are composed of the three degrees of freedom equations of translation and the pseudo-two degrees of freedom approximating the rotational dynamics. The polar coordinates version of the translation equations is used and the controlled rotational dynamics are approximated by two first-order transfer functions (one for the in-plane motion and another for the out-of-plane motion). A time constant of 1 s for both transfer functions is coherent with the requirements of an exo-atmospheric control function [17]. Combining these two sets of equations gives the complete nonlinear pseudo-5DoF equations:

$$\dot{m} = \Delta m \tag{1a}$$

$$\dot{r} = v\,\sin(\gamma) \tag{1b}$$

$$\dot{v} = \frac{T\,\cos\,\vartheta\,\cos\,\varphi}{m} - g_r\,\sin\,\gamma + g_\delta\,\cos\,\chi\,\cos\,\gamma - \omega_e^2 r\,\cos\,\delta(\sin\,\delta\,\cos\,\chi\,\cos\,\gamma - \cos\,\delta\,\sin\,\gamma) \tag{1c}$$

$$\dot{\delta} = \frac{v\,\cos\,\gamma\,\cos\,\chi}{r} \tag{1d}$$

$$\dot{\lambda} = \frac{v\,\cos\,\gamma\,\sin\,\chi}{r\,\cos\,\delta} \tag{1e}$$

$$\dot{\chi} = \frac{T\,\cos\,\vartheta\,\sin\,\varphi}{mv\,\cos\,\gamma} + \frac{v}{r}\,\sin\,\chi\,\cos\,\gamma\,\tan\,\delta - \frac{g_\delta\,\sin\,\chi}{v\,\cos\,\gamma} + \frac{\omega_e^2}{v\,\cos\,\gamma} r\,\cos\,\delta\,\sin\,\delta\,\sin\,\chi - \frac{2\omega_e}{\cos\,\gamma}(\sin\,\gamma\,\cos\,\delta\,\cos\,\chi - \cos\,\gamma\,\sin\,\delta) \tag{1f}$$

$$\dot{\gamma} = \frac{-T\,\sin\,\vartheta}{mv} - \frac{g_r}{v}\,\cos\,\gamma - \frac{g_\delta}{v}\,\cos\,\chi\,\sin\,\gamma + \frac{v}{r}\,\cos\,\gamma + 2\omega_e\,\sin\,\chi\,\cos\,\delta + \frac{\omega_e^2 r\,\cos\,\delta}{v}(\cos\,\gamma\,\cos\,\delta + \sin\,\delta\,\cos\,\chi\,\sin\,\gamma) \tag{1g}$$

$$\dot{\vartheta} = \vartheta_{com} - \vartheta \tag{1h}$$

$$\dot{\varphi} = \varphi_{com} - \varphi \tag{1i}$$

where m is the launcher mass. The instantaneous mass consumption (Δm) and the thrust magnitude (T) are motor-related constants. The r and v variables are respectively the magnitude of the radius (distance between the Earth center and launcher center of mass) and speed vectors. The latitude (δ), the longitude (λ), the flight-path heading (χ) and the flight-path inclination (γ) are angles describing the orientation of the radius and speed vectors. The radius vector and its corresponding angles are presented in Fig. 1 while the speed vector is presented in Fig. 2. Angles ϑ and φ are the in-plane angle and out-of-plane thrust orientation angles respectively (Fig. 2). This representation has discontinuities over the poles ($\delta = \pm 90^{\circ}$) but they are not harmful to the sun-synchronous orbit studied later in this work (Section 4).

In these equations, the variables g_r and g_δ, representing the gravitational acceleration, set the model of the Earth. Most of space launcher guidance functions are designed with either a uniform gravity over a flat Earth or a gravity proportional to the inverse of the square of the radius over a spherical Earth [2]. That being said, launches for high inclination orbits, like the orbit studied as part of this work, are expected to reach high latitudes where the Earth's ellipticity has a significant impact on the gravity [18]. Its inclusion in the guidance model can improve the precision of the resulting trajectory [19]. Hence, the second zonal harmonic is considered in this work. The corresponding gravity approximation is

$$g_r = \frac{\mu_e}{r^2}\left(1 + 1.5J_2\left(\frac{r_{eq}}{r}\right)^2\left(1 - 3\,\sin^2\delta\right)\right) \tag{2}$$

$$g_\delta = \frac{\mu_e}{r^2}\left(3J_2\left(\frac{r_{eq}}{r}\right)^2\,\sin\delta\,\cos\delta\right) \tag{3}$$

Eq. (1) are the state equations of the nonlinear model. A second set of equations, the output equations, is needed to complete it. Instead of tracking the speed and radius vectors as path reference methods usually do, this work suggests tracking the orbital parameters. This choice is mainly justified by the possibility of partially defining the trajectory. The injection anomaly, namely the position of the launcher on its orbit during the injection, is irrelevant for a launcher. In this work, the desired orbit is circular so the argument of perigee is undefined and is unnecessary to the trajectory definition. Moreover, the right ascension of the ascending node is also omitted but, if necessary, it could easily be added to the output vector. Only three parameters of the six related to the components of the radius and speed vectors are then studied. This would not be possible for a vector tracking algorithm as the six components are needed to define the semi-major axis, therefore, fixing the five other orbital parameters. That being said, other values of the vectors components can give the desired semi-major axis but a different value of an unconsidered orbital parameter. The size of the orbit is normally defined by geometric constants: semi-major axis (a) and eccentricity (e):

$$a = \frac{r\mu_e}{2\mu_e - rv_I^2} \tag{4}$$

$$e = \sqrt{\left(\frac{v_I^2 r}{\mu_e}\right)^2\cos^2\gamma_I - 2\frac{v_I^2 r}{\mu_e}\cos^2\gamma_I + 1} \tag{5}$$

The subscript I refers to the inertial values of the parameters based on the inertial speed vector ($\boldsymbol{v_I} = \boldsymbol{v} + \boldsymbol{\omega_e} \times \boldsymbol{r}$). Inertial values are needed because the orbital parameters are defined in an inertial frame whereas the speed vector of Eq. (1) is the non-inertial flight-path speed. That being said, numerical problems resulting from an eccentricity close to zero, arise in the computation of the linear varying models and in MPC optimization algorithms. Therefore, this work suggests defining the orbit with its dynamic constants: the specific angular momentum (h) and the specific mechanical energy ($\mathcal{E}$):

$$h = rv_I^2\,\cos\gamma_I \tag{6a}$$

$$\mathcal{E} = \frac{v_I^2}{2} - \frac{\mu_e}{r} \tag{6b}$$

The derivatives, inside the time-varying representations, of both of these expressions exist and give finite values. Furthermore, both values are far from being null, and this will help in the MPC algorithms.

The orbital inclination is defined by the angle between the orbital plane and equatorial plane:

$$i = \arccos(\sin\chi_I\,\cos\delta) \tag{6c}$$

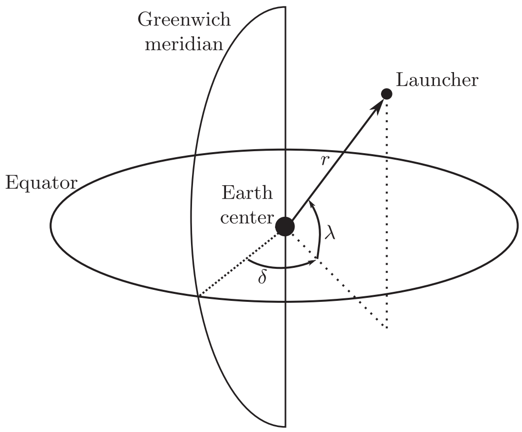

Fig. 1. Radius vector definition.

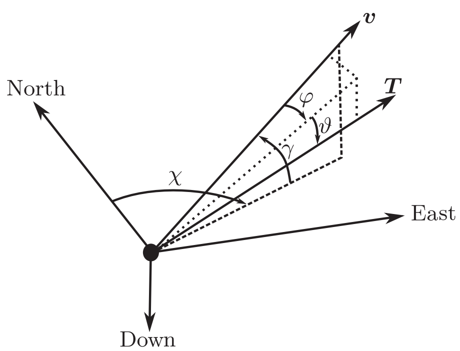

Fig. 2. Speed and thrust vectors definition.

Eqs. (1) and (6) are the NMPC predictive model and the bases of the linear time varying representations. These time-varying representations are interpolations of a family of linear models obtained from a referential trajectory. Each linear model is an evaluation of the linear tangent model. These representations, LTVR and LFR, are presented in the next two sections.

2.2. Linear-time-varying representation

Time is the scheduling parameter of the linear-time-varying representation. For launcher guidance, using time as the scheduling parameter is not problematic as the launcher is expected to follow a predefined trajectory closely where time is a meaningful parameter. Time affects the launch trajectory because orbital parameters are linked directly to it. Furthermore, for the application in a MMPC algorithm (Section 3), time must be measured or estimated as it defines the referential trajectory. Time evolution is also known *a priori*, which simplifies and accelerates optimizations.

LTVR consists in representing the nonlinear process (Eqs. (1) and (6)) as a model with the following form:

$$\Delta_o \dot{\boldsymbol{x}} = \mathbf{A_e}(t)\Delta_o \boldsymbol{x} + \mathbf{B_e}(t)\Delta_o \boldsymbol{u} \tag{7}$$

$$\Delta_o \boldsymbol{y} = \mathbf{C_e}(t)\Delta_o \boldsymbol{x} \tag{8}$$

where the state vector ($\boldsymbol{x}$) is $[m\ r\ v\ \delta\ \lambda\ \chi\ \gamma\ \vartheta\ \varphi]^T$, the input vector ($\boldsymbol{u}$) is the commanded thrust orientation angles ($[\vartheta_{com}\ \varphi_{com}]^T$) and the output vector ($\boldsymbol{y}$) contains the studied orbital parameters ($[h\ \mathcal{E}\ i]^T$). The notation Δ_o stands for variations around the operating points $\boldsymbol{\Delta_o u} = \boldsymbol{u} - \boldsymbol{u}_{op}(t)$.

The matrices $\mathbf{A_e}(t)$, $\mathbf{B_e}(t)$ and $\mathbf{C_e}(t)$ are the interpolations of a family of linear models. They consist of polynomial expressions that best suit the family of linear models. Given the large differences between the values inside the matrices, polynomial fitting is completed separately on each element of the matrices. For the current application, a fifth order polynomial is suitable. Hence, for example, the polynomial expression of the element located in the first column and first row of matrix $\mathbf{A_e}$ (A_{e11}) is obtained by solving the following expression of a least squares problem:

$$\begin{bmatrix} t^{(1)^5} & t^{(1)^4} & t^{(1)^3} & t^{(1)^2} & t^{(1)} & 1 \\ t^{(2)^5} & t^{(2)^4} & t^{(2)^3} & t^{(2)^2} & t^{(2)} & 1 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ t^{(m)^5} & t^{(m)^4} & t^{(m)^3} & t^{(m)^2} & t^{(m)} & 1 \end{bmatrix} \begin{bmatrix} A_{e11}^5 \\ A_{e11}^4 \\ A_{e11}^3 \\ A_{e11}^2 \\ A_{e11}^1 \\ A_{e11}^0 \end{bmatrix} = \begin{bmatrix} A_{11}^{(1)} \\ A_{11}^{(2)} \\ \vdots \\ A_{11}^{(m))} \end{bmatrix} \tag{9}$$

where $t^{(n)}$ is the value of t at the nth operating point. This equation is written and solved, using the pseudo-inverse [20], for each element of matrices $\mathbf{A}$ and $\mathbf{C}$. Matrix $\mathbf{B}$ is constant.

The matrix $\mathbf{A_e}(t)$ in the Eq. (7) is then obtained by evaluating the resulting polynomial expressions:

$$\mathbf{A_e}(t) = \mathbf{A_e^0} + \mathbf{A_e^1}t + \cdots + \mathbf{A_e^5}t^5 \tag{10}$$

where the matrix $\mathbf{A_e^0}$ is

$$\mathbf{A_e^0} = \begin{bmatrix} A_{e11}^0 & A_{e12}^0 & \cdots & A_{e19}^0 \\ A_{e21}^0 & A_{e22}^0 & \cdots & A_{e29}^0 \\ \vdots & \vdots & \ddots & \vdots \\ A_{e91}^0 & A_{e92}^0 & \cdots & A_{e99}^0 \end{bmatrix} \tag{11}$$

2.3. Linear-fractional-representation

Linear-fractional-representation [21] is a special representation of a linear-parameter-varying (LPV) system where parameter dependency is encapsulated in the Δ block of the linear-fractional transformation (LFT) [22]. Hence, the process of obtaining a LFR is similar to the previous section. Matrices $\mathbf{A_e}(t)$, $\mathbf{B_e}(t)$ and $\mathbf{C_e}(t)$ are obtained by fitting the family of linear models. Then, using the generalized Morton's realization technique [23], Eq. (9) is converted in the $M-\Delta$ form. Polynomial coefficients being rational functions of the scheduling parameters, the generalized Morton's technique is needed. Hence, having the minimal order LFR is no longer a guarantee. The matrices of Eqs. (7) and (8) are obtained by computing the resulting upper-LFT [22]:

$$\begin{bmatrix} \mathbf{A_e}(t) & \mathbf{B_e}(t) \\ \mathbf{C_e}(t) & \mathbf{0} \end{bmatrix} = \mathbf{M_{22}} + t\mathbf{M_{21}}\mathbf{I}(\mathbf{I} - t\mathbf{M_{11}}\mathbf{I})^{-1}\mathbf{M_{21}} \tag{12}$$

LFR matrices $\mathbf{M_{11}}$, $\mathbf{M_{12}}$, $\mathbf{M_{21}}$ and $\mathbf{M_{22}}$ result from a least squares problem. Identity matrix ($\mathbf{I}$) has the same dimensions as the order of the LFR. Computing the LFR requires inversing a matrix in which dimension is the same as the order of the realization, thus the importance given to the order of the representation. The two representations (LTVR and LFR) have been applied and validated for the launch ascent trajectory [11].

3. Model predictive control

Model predictive control (MPC) was introduced to the control theory by the petro-chemical industry in the late 1970s [24]. It is a mix between control theory and optimization where an optimization problem is defined based on the control requirements to be met. A model of the system calculates the predicted future outputs ($\hat{\boldsymbol{y}}$) in response to a series of inputs ($\boldsymbol{u}$). Based on the minimization of a cost function that includes the difference between the desired outputs ($\boldsymbol{d}$) and these predicted future outputs, the optimization algorithm finds the optimal inputs to be applied to the process. Depending on the predicting equations, the resulting MPC controller can be a linear MPC, a NMPC or MMPC. The optimal values are then sent to the system and, at the next time step, the optimization starts over using the new measurements. The prediction horizon then keeps being shifted forward. This is the receding horizon principle. Fig. 3 shows an example of a discrete MPC applied on a continuous linear system.

As MPC formulation results in an optimization problem, constraints can be included in the definition of the controller. With constraints, physical limitations of the vehicle

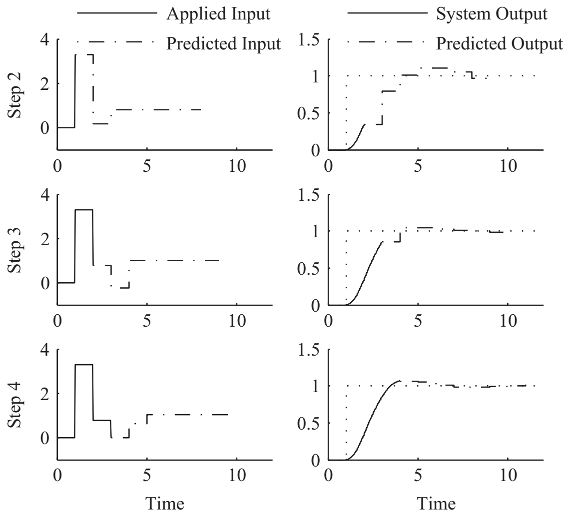

Fig. 3. Schematics of receding horizon principle.

can be addressed and so it can then be operated closer to its optimal performance [25]. These constraints can also define a terminal region ensuring MPC stability [26] or be used to define a shape for the input vector to diminish computation time [27].

That being said, the ability to handle constraints is also a drawback of MPC. It requires the resolution of a constrained optimization problem where convergence inside limited available time is not ensured, leading to suboptimal or diverging solutions. This is why MPC is currently more common in the process industry where settling times are longer [25] and the optimization algorithm has more time to converge.

3.1. Formulation for launch ascent trajectory tracking

The current section formulates the MPC algorithm for the launch ascent guidance problem. The MPC criterion is the differences between the desired and predicted outputs. These outputs, the orbital parameters, define the trajectory, and the MPC becomes a path reference method where the launcher tries to follow a predefined trajectory. A term to limit drastic changes between consecutive inputs is also introduced in the criterion:

$$J = \sum_{l=1}^{h_p} (\boldsymbol{d}_{k+l} - \hat{\boldsymbol{y}}_{k+l})^T \mathbf{Q} (\boldsymbol{d}_{k+l} - \hat{\boldsymbol{y}}_{k+l}) + \sum_{l=1}^{h_c} \boldsymbol{\Delta u}_{k+l-1}^T \mathbf{R} \boldsymbol{\Delta u}_{k+l-1} \tag{13}$$

where the predicted model outputs (the controlled variables) are the orbital parameters: $\hat{\boldsymbol{y}} = [h\ \mathcal{E}\ i]^T$ and the model inputs (the manipulated variables) are the thrust orientation angles $\boldsymbol{u} = [\vartheta_{com}\ \varphi_{com}]^T$. Vector $\boldsymbol{\Delta u}$ is the inputs increment vector, $\boldsymbol{\Delta u}_{k+l} = \boldsymbol{u}_{k+1} - \boldsymbol{u}_{k+l-1}$. MPC produces a combination of inputs separated in time that results in motions that are instantaneously unfeasible and makes the control of underactuated or non-holonomic systems possible with MPC controllers [28]. Prediction and control horizons, respectively, h_p and h_c, and the weighting matrices $\mathbf{Q}$ and $\mathbf{R}$ are design parameters of the guidance law. The value of each parameter depends on the studied trajectory and is explained later in Section 4.

The MPC formulation also includes constraints. With a prediction horizon long enough to always include injection time, terminal orbital parameters are constants and can define constraints ensuring that the orbit reached is the desired one. Furthermore, using inequality constraints combined with a precision vector ($\boldsymbol{\epsilon}_k$) allows for the minimal accepted precision on the orbit to be explicitly specified in the guidance law initialization. This precision is defined in terms of altitude and inclination. Therefore, the apsides radius (r_a and r_p) are more relevant than the dynamics constants (h and $\mathcal{E}$) for constraints defining orbit size:

$$-\boldsymbol{\epsilon}_k \le \begin{bmatrix} r_{a_{k+h_p}} \\ r_{p_{k+h_p}} \\ i_{k+h_p} \end{bmatrix} - \begin{bmatrix} r_{a_d} \\ r_{p_d} \\ i_d \end{bmatrix} \le \boldsymbol{\epsilon}_k \tag{14}$$

From the dynamics constants, apsides radius are calculated as follows:

$$r_a = \frac{-\mu_e}{2\varepsilon}\left(1 + \frac{\sqrt{-2\varepsilon(1-h^2)}}{\mu_e^2}\right) \tag{15}$$

$$r_p = \frac{-\mu_e}{2\varepsilon}\left(1 - \frac{\sqrt{-2\varepsilon(1-h^2)}}{\mu_e^2}\right) \tag{16}$$

Input shaping constraints are also implemented by constraining $\boldsymbol{\Delta u}_{k+l}$ at some specific index l. These constraints give a predetermined shape to the input vector resulting in a reduction of the number of unknowns of the optimization algorithm and its computational time. Again, these are design parameters and their values will be discussed in Section 4.

Other important topics to study are the availability of the state vector components and the ability of the guidance law to eliminate steady-state errors. In a complete guidance-navigation and control scheme, both topics are handled by the navigation function. As stated by Shi et al. [29], the navigation function acts as an observer that eliminates bias from the position and speed vectors and hence eliminates steady-state errors. Furthermore, with this observer, all components become estimated and available to the guidance law.

3.2. Linear multi-model predictive control

The last element defining a MPC formulation is its predictive model, which predicts the future behaviour of outputs. This model includes state equations and output equations. In the case of a launch ascent trajectory tracking, state equations are Eq. (1) while (2) stands for the model outputs. As those equations are nonlinear, the previous formulation is a NMPC. This NMPC formulation was successfully applied to launch ascent trajectory using these nonlinear equations [10]. However, an extension to simplify the optimization problem and accelerate its resolution may be an interesting alternative. An easy way

to do it is to use a linear model instead of a nonlinear model. That being said, the launcher's operating range makes the single linear model unreliable for the whole duration of the launch, but the variation can be expressed as a LTVR [11]. The idea of the proposed formulation is to modify the NMPC by converting the nonlinear equations to LTVR (Section 2.2) or LFR (Section 2.3) in order to obtain a MMPC with pre-identified models. Furthermore, as the models are pre-identified and the scheduling parameters are known *a priori* the minmax optimization of a MMPC predicting around multiple operating regimes like Lu and Arkun [13] is converted into a single minimization. Therefore, nonlinear model predictive control is a special case of MPC in which the predicting equations are nonlinear and MMPC is a also a special case of MPC in which the equations are either the LTVR or the LFR.

3.3. Stability issues of MPC

A closed-loop system is stable at a set point if, when starting out near this point, the system stays close to this set point forever [30]. In the case of launcher guidance, the set point corresponds to the desired orbit, and the stability of the system at this orbit ensures its sustainability. MPC stability resides in problem feasibility and cost function monotonicity [30]. The previous formulation of the problem is somewhat similar to the MPC dual mode control. This formulation includes a constant ellipsoid around the set point, defined by constraints on $\hat{\boldsymbol{y}}_{k+h_p}$, to define two operation modes. Outside this ellipsoid, an MPC controller is employed and, once inside the ellipsoid, the control is switched to a static linear state feedback. As demonstrated by Kwon and Han [30] for linear systems and by Grüne and Pannek [26] for nonlinear cases, this dual-mode formulation ensures the cost monotonicity condition. de Nicolao et al. [31] proved the MMPC stability with constant terminal region. In the current formulation, with $\boldsymbol{\epsilon}_k$ in the constraints, the terminal region varies and the dynamics of the system inside also vary. That being said, if stability using static linear state feedback is possible inside the largest region, MMPC stability is also possible. Dynamic variations are known and bounded; it is then possible to verify that a static linear feedback is robust to these variations ensuring worst-case stability. However, these stability proofs are based on the hypothesis that the predictive model is a perfect representation of the system dynamics, which is not the case for the current application.

Furthermore, even if cost monotonicity can be proven, the finite prediction, control horizons and additional constraints makes satisfaction of the terminal constraints more difficult to guarantee [28]. The problem feasibility condition, the second requirement to guarantee stability, cannot therefore be guaranteed; hence, neither can the MPC stability.

4. Simulations and results analysis

NMPC and MMPC trajectory tracking laws defined in the previous section are applied to the exo-atmospheric trajectory of a three-stages-to-orbit launcher to reach a circular sun-synchronous orbit with a semi-major axis of 6871 km and an orbital inclination of 97.375°. Before applying both MPC formulations, the referential trajectory is obtained by solving an off-line trajectory generation algorithm [19]. The trajectory obtained is a burn-coast-burn one where the coast arc occurs immediately after the second-stage jettison.

With the separation of the trajectory by the coast phase, the size of the NMPC and MMPC formulations can be diminished. During the coast phase, the launcher is unpropelled and the orbit cannot be modified. Therefore, the complete trajectory of the launcher is divided into two distinct guided phases and MPC problems. During the second-stage burn, the guidance law tries to inject the third stage and the attached payload on the coast orbit, calculated by the off-line algorithm. When the third stage is active, the objective of the guidance law is to put the payload on the desired orbit for its mission (Fig. 4).

With this division into two consecutive phases, some of the parameters differ whether the second or third stage is active. These parameters are the prediction and control horizons and the terminal region. During the second-stage burn, the terminal region is defined around the coast orbit and around the desired orbit for the third-stage burn. As the constraints (Eq. (14)) are defined at the end of the prediction horizon, the latter must be long enough to always include the injection time. For an undivided trajectory, the minimal duration of the prediction would be 508.2 s, while for the divided trajectory, the minimal durations are respectively 48.2 and 104.4 s for the pre-coast and the post-coast problems. In MPC, when the control horizon is shorter than the prediction horizon, all input increments after the control horizon are set to zero. That being said, as demonstrated by the off-line algorithm [19], the optimal thrust orientation needed to reach the orbit changes all the way up to the injection. To reproduce this optimal shape, the algorithm must therefore be able to modify the thrust orientation for the whole prediction horizon. Hence, the control horizon must be as long as the prediction horizon. A shorter control horizon means fewer unknowns for the optimization problem and should translate into a shorter computational time. For the current application, with a 1 s guidance period, both the prediction and control horizons are 49 for the pre-coast burn and 105 steps for the post-coast phase as opposed to the 509 for the undivided trajectory. The pre-coast burn phase starts with the fairing jettison, 170.8 s after the launch, and ends at the beginning of the coast phase (219 s). As for the post-coast phase, it starts at the end of the coast phase (569.4 s)

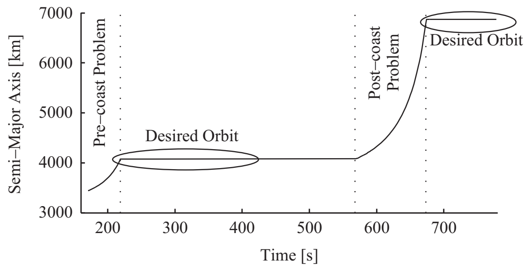

Fig. 4. Launch trajectory separation in two consecutive MPC problems.

and ends with the injection of the payload on its final orbit (673.9 s).

As the order of magnitude of the output vector is similar in both phases, weighing matrices can be the same for the two phases. The values of these diagonal matrices (**Q** and **R**) eliminate problems resulting from different orders of magnitude inside the criterion. The implicit unit of each element gives a good idea of the order of magnitude of the matrix components. Hence, the element of **Q** related to the specific angular momentum should be low to bring the three orbital parameters to the same order of magnitude. These matrices set the relative importance of each parameter. Therefore, as they are more fuel consuming, out-of-plane maneuvers should be hardly penalized. This is done by setting a bigger weight to the out-of-plane angles and the orbital inclination. The element of **R** related to the out-of-plane angles (R_{22}) and the parameter of **Q** related to the orbit inclination (Q_{33}) should be higher than when the matrices serve only for normalization. Preliminary tests showed that the following matrices **Q** and **R** yield good results:

$$\mathbf{Q}=\begin{bmatrix}1\times10^{-6} & 0 & 0\\ 0 & 1\times10^{2} & 0\\ 0 & 0 & 1\times10^{3}\end{bmatrix} \tag{17}$$

$$\mathbf{R}=\begin{bmatrix}1\times10^{3} & 0\\ 0 & 1\times10^{5}\end{bmatrix} \tag{18}$$

Concerning the constraints, the desired orbital parameters are set to either the coast orbital parameters or the injection orbital parameters depending on the phase. Given that they try to counteract the disturbances in a short amount of time, guidance algorithms which do not handle thrust cut-off and request near-perfect injection result in large maneuvers close to injection [32]. For the current formulations, this problematic behavior can be avoided by implementing a time-varying precision vector (ϵ_k). Earlier in the launch, it is set at

$$\epsilon_0=[1\times10^{-1}\ \ 1\times10^{-1}\ \ 1\times10^{-3}]^T \tag{19}$$

to request a near-perfect launch and, as the remaining time before injection decreases, it constantly increases to reach the desired precision for each orbital parameter at injection time, which, for the current application, is set at

$$\epsilon_{h_p}=[4\ \ 4\ \ 1\times10^{-2}]^T \tag{20}$$

The implemented time-varying scheme for the precision is then

$$\epsilon_k=\begin{cases}\epsilon_0 & \text{if } k\le h_p-25\\ \boldsymbol{\epsilon_{var}} & \text{if } h_p-25<k\le h_p-5\\ \epsilon_{h_p} & \text{if } k>h_p-5\end{cases} \tag{21}$$

where $\boldsymbol{\epsilon_{var}}$ is the equation of the increasing slope

$$\epsilon_{var}=\frac{\boldsymbol{\epsilon_{h_p}}-\boldsymbol{\epsilon_0}}{20}(k-20h_p)+25\boldsymbol{\epsilon_{h_p}}-5\boldsymbol{\epsilon_0} \tag{22}$$

Input shaping constraints also give a predetermined shape to the input increments, which translates into a reduction in the number of unknowns in the optimization. This shape consists of free input increments at the beginning of the horizon followed by sequences of one free input increment and three input increments constrained to be null. In the pre-coast phase, the tested shape is composed of 9 free increments followed by 10 sequences (Fig. 5). As for the shape of the post-coast phase, it consists of 41 free inputs and 16 sequences (Fig. 6).

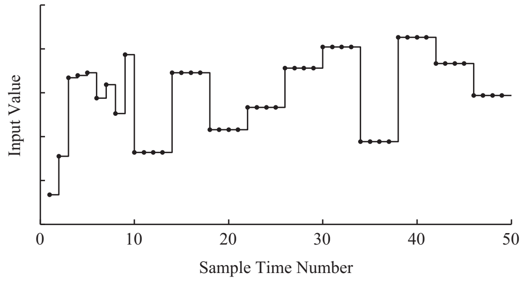

Fig. 5. Example of a constrained shape for the pre-coast problem.

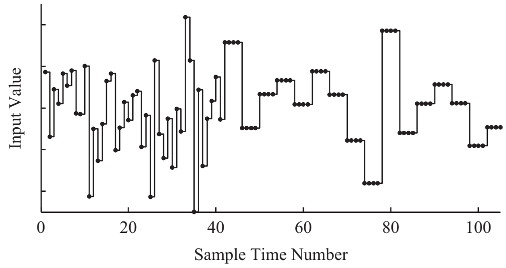

Fig. 6. Example of a constrained shape for the post-coast problem.

Guidance laws are applied to a launch simulator. In the first test (Section 4.1), only random noises corresponding to the output precision of an integrated GPS/INS navigation function are present. Based on the results of Maki [33], random white noises with a covariance of 35 m on the position and 0.1 m/s on the velocity are reasonable estimates of an achievable precision for a space launcher navigation system. This test is conducted to investigate guidance law performances in nominal operating conditions. As the proposed guidance laws are path reference methods, they should perform well in this case [2]. The second series of tests (Section 4.2) add motor uncertainties to the random noises of the first case. The uncertainties are modeled by a difference between the thrust magnitude of the simulator and the one of the predictive model of the guidance laws. For the two series of tests, the proposed MMPCs using the LTVR (LTVR-MPC) and the LFR (LFR-MPC), are compared to the NMPC algorithm [10] modified to handle the time-varying precision and to a specific predictor/corrector method [3].

4.1. Simulation with random noises only

Fig. 7 presents the applied commanded thrust orientations obtained with the four algorithms applied to a simulation with only random noises as perturbations. Tables 1 and 2 present the orbits reached at the end of the two phases.

As expected, in nominal operating conditions, path reference algorithms based perform better than the comparison

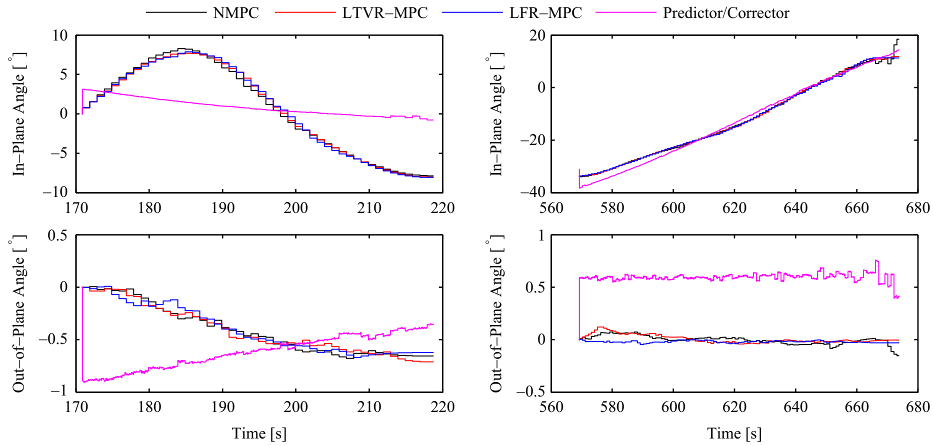

Fig. 7. Applied thrust orientations obtained for random noises only.

Table 1
Reached coast orbits for random noises only.

Guidance method	**Apogee radius** (km)	**Perigee radius** (km)	**Orbital inclination** (deg)
Desired	6862.5	1294.0	97.232
NMPC	6862.6	1294.9	97.233
LTVR-MPC	6862.5	1295.0	97.233
LFR-MPC	6862.5	1294.8	97.230
Predictor/corrector	6861.2	1304.2	97.308

Table 2
Reached final orbits for random noises only.

Guidance method	**Apogee radius** (km)	**Perigee radius** (km)	**Orbital inclination** (deg)
Desired	6871.0	6871.0	97.375
NMPC	6872.9	6869.9	97.376
LTVR-MPC	6875.0	6867.7	97.375
LFR-MPC	6872.4	6870.3	97.376
Predictor/corrector	6871.3	6867.3	97.375

predictor/corrector method. Commanded thrust angles are smaller and the third stage duration is shorter. This translates into a maximal payload mass of 187.5 kg for the MPC algorithms and 186.2 kg for the comparison predictor/corrector method. The orbit reached by the three MPC algorithms are, at worst, as precise as the referential method. There are no significant differences between the LFR, LTVR and non-linear model [11]; this is also visible in Fig. 7. The angles obtained are nearly identical for the three algorithms. That being said, the investigation of computation time (Fig. 8) draws a completely different conclusion.

Even if it would be logical to think that path reference algorithms should have a better computation time, MPC formulations are worse than the comparison predictor/corrector method by far. This can be explained by the MPC formulation. It results in a constrained optimization that is not easier to solve than the predictor/corrector method. The complexity of MPC formulations explains why the predictor/corrector method, which uses hypothesis to simplify the optimization problem, is quicker. The difference between the three MPCs is explained by the complexity of predictive models. The LTVR is the easiest to compute, the LFR, by the matrix inversion, is longer and the NMPC is the longest.

4.2. Simulation with random noises and motors misfunction

In this section, multiple tests were conducted to verify the robustness of the developed guidance laws to a model mismatch. To do so, the thrust of the real launcher was modified. The main results of these tests are the inefficiency of the NMPC and, particularly the MMPCs. When the malfunction is relatively small, roughly below 3% for less than 40 s, the four algorithms are almost unaffected by the model error; however, when the difference is larger, the MMPCs have trouble providing the desired orbit. Table 3 presents the final orbit achieved when the third stage thrust is 3% lower than its nominal value for its whole duration and Table 4 shows the results when a 3% malfunction occurs 640 s after the launch and lasts until the injection.

These results can be explained by the fact that the linear MMPC approximations are valid only around the referential trajectory for which they have been defined. That being said, a malfunction steers them away from this trajectory. The validity of the approximation is gradually lost, decreasing performances. As for the NMPC, it is not defined around a single trajectory and is valid for a wider range. So it does converge to the desired orbit; however, it requires large maneuvers to do so because it tries to follow a trajectory that cannot be attained with the launcher's new configuration. The comparison predictor/corrector method seems to be less sensitive to motor malfunctions; its result is relatively unaffected by a change in the thrust magnitude. This is explained by the fact that the design of this method is based on a highly simplified model. Hence, the algorithm must be robust to model uncertainties even in nominal conditions. These results are coherent with the preference for the predictor/corrector

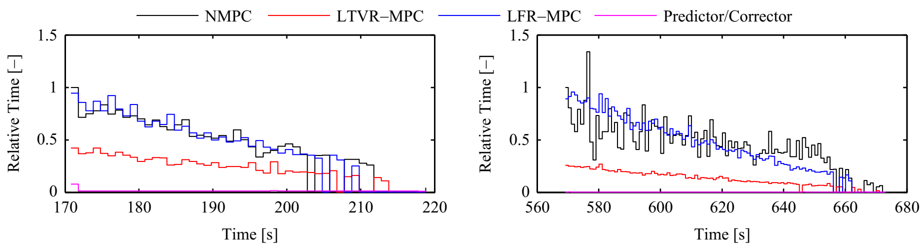

Fig. 8. Relative computation time needed for random noises only.

Table 3
Reached final orbits for random noises and a 3% thrust difference for 105 s.

Guidance method	Apogee radius (km)	Perigee radius (km)	Orbital inclination (deg)
Desired	6871.0	6871.0	97.375
NMPC	6868.0	6863.9	97.369
LTVR-MPC	6871.9	6268.2	97.363
LFR-MPC	6870.1	6186.7	97.358
Predictor/corrector	6871.1	6865.2	97.374

Table 4
Reached final orbits for random noises and a 3% thrust difference for 34.9 s.

Guidance method	Apogee radius (km)	Perigee radius (km)	Orbital inclination (deg)
Desired	6871.0	6871.0	97.375
NMPC	6872.2	6870.8	97.376
LTVR-MPC	6874.4	6868.7	97.375
LFR-MPC	6873.8	6868.5	97.375
Predictor/corrector	6871.8	6869.8	97.375

methods in space launcher guidance because they perform better under off-nominal conditions [9].

5. Conclusion

This work has developed trajectory tracking guidance laws based on discrete model predictive control. A first algorithm is based on a nonlinear model of the translation dynamics, and the others use a time-varying approximation of the nonlinear translation dynamics as their predictive model. Tracking is done using the orbital parameters as outputs; this enables a partial definition of the orbit. The orbital parameters define an invariant terminal zone using constraints on the last prediction. This is similar to the algorithm in the dual mode MPC, which ensures stability and makes it possible to include the desired precision directly in the definition of the guidance law. Moreover, to reduce the number of unknowns in the optimization algorithm, the burn-coast-burn trajectory is divided into two separate guidance problems, and input shaping constraints are implemented. Multi-model predictive control algorithms prove to be a viable alternative to nonlinear model predictive control for a nominal launch, particularly the linear-time-varying representation, which is significantly quicker. That being said, its computation time still underperforms a predictor/corrector method used for comparison. Furthermore, from the simulation results, it seems that to become a truly viable solution for a space launcher exo-atmospheric guidance law, the multi-model predictive control algorithms probably still need to have their robustness to input uncertainties improved.

Acknowledgments

The authors would like to thank Numérica Technologies Inc., Natural Sciences and Engineering Research Council of Canada (NSERC) and Fonds de recherche du Québec – Nature et technologies (FQRNT) for their financial support, which made this work possible. We would also like to thank Defence Research and Development Canada (DRDC) for initiating and supporting this project.

References

[1] P. Teofilatto, E. de Pasquale, A fast guidance algorithm for an autonomous navigation system, Planet. Space Sci. 46 (11–12) (1998) 1627–1632, http://dx.doi.org/10.1016/S0032-0633(97)00233-X.
[2] S.K. Shrivastava, M. Seetharama Bhat, S.K. Sinha, Closed loop guidance of satellite launch vehicle – an overview, J. Inst. Eng. (India): Aerosp. Eng. J. 66 (2) (1986) 62–76.
[3] G.A. Dukeman, Closed-loop nominal and abort atmospheric ascent guidance for rocket-powered launch vehicles (Ph.D. thesis), Georgia Institute of Technology, May 2005.
[4] S.K. Sinha, S.K. Shrivastava, Optimal explicit guidance for three-dimensional launch trajectory, Acta Astronaut. 19 (6–7) (1989) 555–560, http://dx.doi.org/10.1016/0094-5765(89)90123-9.
[5] A. Mohammadi, J. Roshanian, M. Bahrami, B. Ebrahimi, On application of q-guidance method for satellite launch systems, in: Proceedings of the 3rd International Symposium on Systems and Control in Aeronautics and Astronautics, Institute of Electrical and Electronics Engineers, Harbin, China, 2010, pp. 1314–1319, http://dx.doi.org/10.1109/ISSCAA.2010.5632295.
[6] P. Lu, Nonlinear trajectory tracking guidance with application to a launch vehicle, J. Guidance, Control Dyn. 19 (1) (1996) 99–106, http://dx.doi.org/10.2514/3.21585.
[7] E.N. Johnson, A.J. Calise, J.E. Corban, Adaptive guidance and control for autonomous launch vehicles, in: Proceedings of the IEEE Aerospace Conference, vol. 6, Institute of Electrical and Electronics Engineers, Big Sky, USA, 2001, pp. 2669–2682, http://dx.doi.org/10.1109/AERO.2001.931288.
[8] C. Tournes, Y. Shtessel, Predictive launcher guidance using second order sliding mode control, in: AIAA Guidance, Navigation and Control Conference, No. AIAA 2006-6799, Keystone, USA, 2006.
[9] S. Su, Advanced guidance algorithms for the ARES V cargo launch (Master's thesis), University of Missouri-Columbia, August 2008.
[10] A. Vachon, A. Desbiens, E. Gagnon, C. Bérard, Space launcher guidance based on discrete nonlinear model predictive control, in: Proceedings of the 63rd International Astronautical Congress,

No. IAC-12-C1.1.6, International Astronautical Federation, Naples, Italy, 2012.
[11] A. Vachon, A. Desbiens, E. Gagnon, C. Bérard, LFR and LTV representation of space launcher translation dynamics, in: Proceedings of the 16th CASI Astronautics Conference, No. AS 12-042, Canadian Aeronautics and Space Institute, Quebec City, Canada, 2012.
[12] M.V. Kothare, B. Mettler, M. Morari, P. Bendotti, C.-M. Falinower, Level control in the steam generator of a nuclear power plant, IEEE Trans. Control Syst. Technol. 8 (1) (2000) 55–69, http://dx.doi.org/10.1109/87.817692.
[13] Y. Lu, Y. Arkun, Quasi-min–max MPC algorithms for LPV systems, Automatica 36 (4) (2000) 527–540, http://dx.doi.org/10.1016/S0005-1098(99)00176-4.
[14] B. Aufderheide, B.W. Bequette, Extension of dynamic matrix control to multiple models, Comput. Chem. Eng. 27 (8–9) (2003) 1079–1096, http://dx.doi.org/10.1016/S0098-1354(03)00038-3.
[15] Z. Xu, J. Zhao, J. Qian, Y. Zhu, Nonlinear MPC using an identified LPV model, Ind. Eng. Chem. Res. 48 (6) (2009) 3043–3051, http://dx.doi.org/10.1021/ie801057q.
[16] P.H. Zipfel, Modelling and Simulations of Aerospace Vehicle Dynamics, 2nd edition, AIAA Education Series, AIAA, Inc., Reston, USA, 2007.
[17] G. Baldesi, Modelling, control design and simulation for a launch vehicle: from linear to nonlinear methods (Ph.D. thesis), Institut Supérieur de l'Aéronautique et de l'Espace, 2008.
[18] E.M. Gaposchkin, K. Lambeck, 1969 Smithsonian standard Earth (ii) (Technical Report) SAO Special Report 315, Smithsonian Astrophysical Observatory, May 1970.
[19] A. Vachon, A. Desbiens, E. Gagnon, C. Bérard, Exo-atmospheric launch trajectory definition using direct optimization and slack variables, Can. Aeronaut. Space J. 59 (01) (2013) 25–33, http://dx.doi.org/10.5589/q13-004.
[20] N.J. Higham, Accuracy and Stability of Numerical Algorithms, 2nd edition, Society for Industrial and Applied Mathematics, Philadelphia, USA, 2002.
[21] J.-F. Magni, User manual of the linear fractional representation toolbox, version 2.0 (Technical Report TR 5/10403.01F DCSD), ONERA, The French Aerospace Lab, February 2006.
[22] K. Zhou, J.C. Doyle, K. Glover, Robust and Optimal Control, Prentice-Hall, Inc., Upper Saddle River, USA, 1996.
[23] Y. Cheng, B. DeMoor, A multidimensional realization algorithm for parametric uncertainty modelling and multiparameter margin problems, Int. J. Control 60 (5) (1994) 789–807, http://dx.doi.org/10.1080/00207179408921495.
[24] C.E. Garcia, D.M. Prett, M. Morari, Model predictive control: theory and practice – a survey, Automatica 25 (1989) 335–348, http://dx.doi.org/10.1016/0005-1098(89)90002-2.
[25] J.M. Maciejowski, Predictive Control with Constraints, Prentice Hall, Essex, England, 2002.
[26] L. Grüne, J. Pannek, Nonlinear Model Predictive Control, Theory and Algorithms, Communications and Control Engineering, Springer-Verlag, London, England, 2011.
[27] U. Halldorsson, M. Fikar, H. Unbehauen, Multirate nonlinear predictive control, in: Proceedings of the 2002 American Control Conference, vol. 5, American Automatic Control Council, Anchorage, USA, 2002, pp. 4191–4196, http://dx.doi.org/10.1109/ACC.2002.1024589.
[28] C.M. Pong, A. Saenz-Otero, D.W. Miller, Autonomous thruster failure recovery on underactuated spacecraft using model predictive control, in: AAS Guidance and Control Conference, No. AAS 11-033, Breckenridge, USA, 2011.
[29] H. Shi, J. Zhu, Z. Sun, INS/GPS integrated system state estimation based on Hopfield neural network, in: International Conference on Neural Networks and Brain, vol. 2, Institute of Electrical and Electronics Engineers, Beijing, China, 2005, pp. 975–979, http://dx.doi.org/10.1109/ICNNB.2005.1614782.
[30] W. Kwon, S. Han, Receding Horizon Control: Model Predictive Control for State Models, Advanced Textbooks in Control and Signal Processing, Springer-Verlag, London, England, 2005.
[31] G. de Nicolao, L. Magni, R. Scattolini, Stabilizing receding-horizon control of nonlinear time-varying systems, IEEE Trans. Autom. Control 43 (7) (1998) 1030–1036, http://dx.doi.org/10.1109/9.701133.
[32] R.V. Vittal, M.S. Bhat, Optimal explicit terminal guidance for a launch vehicle, Acta Astronaut. 29 (4) (1993) 249–262, http://dx.doi.org/10.1016/0094-5765(93)90138-M.
[33] S.C. Maki, A low-cost GPS inertial guidance (GPSIG) for space boosters, in: IEEE Symposium on Position Location and Navigation, Institute of Electrical and Electronics Engineers, Las Vegas, USA, 1990, pp. 176–183, http://dx.doi.org/10.1109/PLANS.1990.66174.